APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME: Basso et al

TITLE: METHOD AND STRUCTURE FOR INDICATION OF LAST DATA BUFFER OF A FRAME PACKET IN A NETWORK PROCESSOR

DOCKET NO. RAL920000114US1 (IRA-10-5550)

INTERNATIONAL BUSINESS MACHINES CORPORATION

METHOD AND STRUCTURE FOR INDICATION OF LAST DATA BUFFER OF A FRAME PACKET IN A NETWORK PROCESSOR

FIELD OF THE INVENTION

This invention relates generally to network processors and, more particularly, to a structure and method for determining when a network processor has completed a frame transmission and, even more particularly, to an efficient utilization of data in a buffer control block for determining the completion of a data transmission in one or more data buffers comprising the frame.

BACKGROUND INFORMATION

When a network processor (NP) transmits frames of information, these frames are generally comprised of a sequence of buffers chained together. Particularly, each buffer contains a space for a predetermined number of bytes of data; for example, a typical number is 64 bytes. This frame constitutes a packet of data to be transmitted as an individual or unitary transmission. The number of individual buffers, typically dynamic RAMS (DRAMs) comprising a packet or frame of information can vary from one to many buffers which have to be chained together. Typically, the chaining together of the necessary data containing buffers is done by buffer control blocks which conventionally are implemented in static RAMS (SRAMs) for speed. However, the bandwidth of static RAMS is limited and this limits the fields that can be used to determine when all of the data containing buffers of the frame has been transmitted. Although the network typically contains a byte count field to count the number of bytes in the frame, this cannot be used for determining the end of a transmission because a frame frequently is altered or modified by the network processor after being received (which is when the byte count is recorded) and before it is transmitted. Recording two byte counts, the original and modified, is costly

and time consuming and resource intensive when using static RAMS for implementing the buffer control block. Moreover, while it is possible that the network processor could examine the frame alteration field within a frame buffer to determine if the length of the frame was altered and, if so, by how much, this results in added latency penalty for high performance network processors that access slow dynamic RAMs and adds complexity of a function to be implemented in hardware. Therefore, it is desirable to provide a convenient technique for determining the end of a frame transmission utilizing the information contained within a buffer control block.

SUMMARY OF THE INVENTION

A method and structure for determining when a frame of information comprised of one or more buffers of data being transmitted in a network processor has completed transmission is provided. The network processor includes a plurality of control blocks, one for each data buffer, each containing control information to link one buffer to another for transmission. Each of the control blocks has a last bit feature which is a single bit and indicates when the data buffer having the last bit is transmitted. This last bit feature is a bit which can be set to either zero or one. The last bit feature is in a first position when an additional data buffer is to be chained to a previous data buffer indicating an additional data buffer is to be transmitted, and a second position when no additional data buffer is to be chained to a previous data buffer. The position of the last bit feature is communicated to the network processor to indicate whether the transmission of a particular frame is ended and a new frame is to be transmitted.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagrammatic representation of a frame comprised of a plurality of data containing buffers showing how the frame buffer control (FBC) and buffer control blocks (BCB's) are associated therewith for control information;

Figure 2 is a diagrammatic representation of buffer control blocks which are in the free queue;

Figure 3 is a diagrammatic representation of the spaces being used in each buffer control block in the free queue;

Figure 4 is a diagrammatic representation of the spaces used in a buffer control block when it is being utilized to control the transmission of information from a frame comprised of one or more buffers of data information; and

Figure 5 is a flow chart of the operation of the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Before describing the invention in detail, a brief overview of the functioning of a network processor as it relates to the environment of the present invention is presented to aid in understanding the invention. In a network processor, in which the present invention is utilized, information is transmitted from a remote originating location to a central location in a network processor and then formatted in frames of information to be retransmitted to one or more remote locations that are the receiving locations. The frames of data information are typically comprised of one or more buffers of data, each of which buffers has a finite, known length, e.g. 64 bytes. The transmitted data may contain information that is to be either deleted or modified prior to transmission. Thus, as indicated above, although a byte count is typically made of the number of bytes received from the remote

location, the number of bytes transmitted to the receiving locations may be a different number. Also, the number of changed bytes, although they may be recorded, requires manipulation to constitute a measure for determining end of transmission. It is necessary for the network processor to "know when" the transmission is ended so that the network processor can decide which frame is next to be transmitted and bring this frame up for transmission.

According to one scheme of transmitting frames of data comprised of finite length data buffers, buffer control blocks are utilized. These buffer control blocks are utilized to control the sequence of transmission of the data in the frame but are not themselves transmitted. These buffer control blocks are maintained typically in SRAM in what is known as a free queue of the buffer control blocks. When a frame of data is received, a buffer control block is taken from the free queue and associated with one buffer of information in the frame. Each buffer of information in the frame is assigned a control block.

A frame control block (FCB) is utilized to initiate the transmission of buffers in the frame having a first buffer address (FBA). This technique is shown diagrammatically in Figure 1 wherein a frame of data 10 is shown which is comprised of a plurality of data buffers 12-1, 12-2, 12-3, 12-4,...12-n. It is to be understood that in some cases but a single data buffer would be utilized, i.e. that is all the space that is required for the data being transmitted. However, in many cases, more than one data buffer is required and, hence, these must be transmitted together sequentially as a unit. Also, the data in each of the buffers must be accurately located as to the starting and stopping position for the

reading of the data and transmission of the read data especially in data buffer 12 which has been modified.

To start the transmission of a frame, a frame control buffer (FCB) 14 is provided and buffer control blocks 16-1, 16-2, 16-3, 16-4,...16-n are provided which are associated with each of the data buffers 12-1 --12-n. The frame control buffer starts the operation of reading the information from the frame 10 by providing a frame buffer address which will indicate which buffer from the free buffer queue list (FQL), as will be described presently, is to be associated with the first data buffer 12-1. The frame control buffer 14 also contains a parity bit, a starting buffer address 22 and an end buffer address 24, which are, respectively, the starting address in the buffer 12-1 and the end buffer address also in buffer 12-1.

The buffer control block buffers in the free buffer queue are shown diagrammatically in Figure 2. The free buffer queue list is no more than an indication of which buffer control blocks 16-1 – 16-n are associated with data buffers 12-1 through 12-n and are not already in use. The various fields of information contained in a buffer in the free buffer queue are shown in Figure 3. As seen in Figure 3, each buffer control block 16 includes a space 30 in which the next buffer address (NBA) is contained, a blank space which is one bit wide 32, the purpose of which will be indicated later, a blank space 33 which is also one bit wide, the purpose of which also will be described later but which is not related to this invention, a space 34 which is one bit wide and which is a flag bit, often referred to as a last bit (LB), to indicate that either the last bit will occur in that buffer or whether more buffers are to be utilized in conjunction with data buffer 12-1 – 12-n, in a way which will be described presently. A blank space 36 and a space 38 are provided,

which space 38 contains error correction code (ECC) to correct the address of the next buffer address if there are any mistakes contained in space 30. As will be described presently, error correction code (ECC) resides in space 38 but only while the control block buffer is in the free buffer queue as shown in Figure 2. The ECC will assure that the proper next buffer address is contained in the buffer control blocks 16-1 -- 16-n. Also, whenever a buffer control block buffer 16 is in the free buffer queue, the last bit or flag bit space 34 is set to a one.

Once the frame control buffer 14 has selected an address for initial data buffer 12-1, the buffer control block 16-1 is assigned to this specific data buffer 12-1. The frame control buffer 14 has stored therein, as indicated earlier, the starting buffer address 22 from the buffer 12-1 that is being used and the end buffer address 24, also from the buffer 12-1 is being used. The buffer control block 16-1 assigned to the data buffer 12-1 then has written therein a parity bit in space 32 and a transient buffer designation in space 33. (This transient buffer is not a part of the invention so it will not be discussed further.)

The starting address or start byte position (SBP) for the next buffer control block 16-2 (which corresponds to the starting address of the next data buffer 12-2) is written in space 36. The end address or end byte position (EBP) for the data and data buffer 12-2 is then written in space 38 (which previously contained the error correction code) and the last bit flag bit in space 34 is flipped from one to zero, indicating that there is data to be read from the data buffer 12-2. This sequence continues through buffer control block 16-2 associated with data buffer 12-2, buffer control block 16-3 associated with data buffer 12-3, buffer control block 16-4 associated with data buffer 12-4 through the last buffer control block 16-n associated with data buffer 12-n.

Let us take as an example a case where there is data contained within data buffers 12-1, 12-2, 12-3 but none in 12-4, the data ending in data buffer 12-3. In such case, the blanks 36 in control block 16-1 would be written with the starting address of the data in data buffer 12-2 and the space 38 would be written with the end address of the data in data buffer 12-2. The space 36 in buffer control block 16-2 would be written with the starting address of the data in data buffer 12-3 and the space 38 would be written with the ending address of the data contained in data buffer 12-3. In each of these cases, the bits in the last bit space 34 would be flipped to zero. However, with respect to the information written in data control block 16-3, since there is no data contained in frame buffer 12-4, no starting address would be listed in space 36 and ending address in space 38, thus indicating that there is indeed no further data buffers containing information and that data buffers 12-1, 12-2 and 12-3 contain all of the information. Since no starting and ending addresses for data buffer 12-4 are found in spaces 36 and 38 of buffer control block 16-3, the space 34 would contain a bit "one" rather than "zero" since it would not be flipped as no addresses were written. Since this buffer control block 16-3 shows a last bit indication of a one, which information is transmitted to the network processor, the processor "knows" that data buffer 12-3 is the last data buffer that has to be transmitted and can then start queuing the next frame for transmission. The buffer control blocks all have their last bits in space 34 flipped and then buffer control blocks 16-1 to 16-3 are returned to the free buffer control block queue.

If there is data only in the first data buffer 12-1, then the buffer control block 16-1 will contain no starting address or finishing address, and the buffer control block will be

returned to the free queue. (The flipping of the last bit control bit is not required since it is already at "one").

Figure 5 shows a flow chart for the operation of the invention, and is self-explanatory.

Certain important features of this invention include an optimized way to indicate the last buffer in a list of chained buffers based on "remembering" that a buffer came from the free queue. A "last bit" is set to "one" in BCB when releasing a buffer, i.e. enqueueing it in the free queue. When a buffer is leased, i.e. dequeued from the free queue, it comes with its "last bit" set to "one" in the BCB. When a leased buffer is chained to the end of a list of buffers in a frame, its "last bit" is not changed from "one" because its BCB is not written (there is no BCB after this last BCB in the list of buffers). The "last bit" of the buffer which was at the end of the list is reset because the BCB is written to point to the new last buffer (the leased one). The set and reset operations on the "last bit" are free in terms of memory bandwidth because they are included in read or write accesses needed to get or update the next BCB pointer when dequeueing or enqueueing.

Accordingly, the preferred embodiment(s) of the present invention has been described. With the foregoing description in mind, however, it is understood that this description is made only by way of example, that the invention is not limited to the particular embodiments described herein, and that various rearrangements, modifications, and substitutions may be implemented without departing from the true spirit of the invention as hereinafter claimed.